THE OPEN UNIVERSITY OF KENYA

DESIGN PLAN

Programme title	Bachelor of Science in Cybersecurity and Digital Forensics
Course title	Operating Systems
Learning Module number	02
Learning module title	Structure and Components of Operating Systems
Module Developer	Elisha Abade
Module duration in hours	8
Instructional Hour Equivalent (Divide duration by 2)	4
Reviewed by	
Vision	The innovative university for inclusive prosperity
Audience description	Learners of Cyber Security in first semester of second year
Instructions to learners ⓘ	In this course we shall be learning about the components that make up an Operating system and how they are organized into a single unit that achieves the objectives of an Operating System.
Learning module description	This module aims to facilitate learners to have an understanding of the structure and components of an Operating Systems.
Module objectives:	This module aims at facilitating learners to acquire knowledge about: 1. The components of an Operating System 2. The organization of the components of an Operating System 3. The structure of an Operating System
Module learning outcomes:	By the end of the module, you should be able to: 1. Describe components of an Operating System 2. Illustrate organization of the components of an Operating System 3. Sketch the structure of an Operating System 4. Evaluate structures of an operating system based that are fit for specific applications
Planned Learning Resources	
ACTIVITY 1: INTRODUCTION VIDEO 1: Pre-recorded lecture on topic emphasizing LEARNING OUTCOME 1: Factual knowledge.	Topic content is fully presented here. The lecture will deal with factual knowledge, expounding on threshold concepts, if any. **Video 1: Components of an Operating System (8 minutes)** **Introduction**

Welcome to the first session of this module on Introduction to Operating Systems. In this session we shall be looking at the various components that make up an Operating system. These are fully functional units within the Operating system but work collaboratively to realize the overall objective of the functions of an Operating System.

Various Operating Systems differ in structure and organizations but all of them have similar components which offer specialized functions. An Operating System can be seen to have at least eight distinct components. These components perform functions that can be inferred from their names as follows:

1. Command Interpreter System Unit

   In order to be useful to users of computer systems, an Operating System must provide mechanisms for communicating with the users. This communication usually takes the form of taking instructions (commands) from the users, acting on them and presenting the results to users.

   The component of the Operating System that is responsible for this kind of interaction is called the **Shell** or **Command Interpreter**.

   In some Operating Systems, this component communicates to users through text messages from the keyboard and displayed on a screen. They are typically referred to as the "**console**" or "**command line prompt**".

   In other systems, this component communicates to users through pictorial representation of objects on a screen as icons. These are referred to as **Graphical User Interfaces (GUI)**. The icons are manipulated through the mouse in a point and click fashion.

   Insert a diagram here that comprises two concentric rectangles. The inner one labeled "Kernel" while the other outer one labeled "Shell". Put arrows from the outer rectangle on all sides pointing at "users"

2. The Kernel:

   This is the "core" or "the heart" of the Operating System. It contains software that performs very fundamental functions of a computer system. These include:

a. Main Memory Management Unit: Responsible for coordination of the usage of the main memory. This is very important in a multitasking environment where the computer does many tasks concurrently and at times these tasks require more memory than the total amount of main memory available hence the use of a concept called virtual memory.

b. Process Management Unit: This further comprises the scheduler and the dispatcher. The scheduler determines the order in which processes are executed. The dispatcher on the other hand is responsible for dispatching processes so that they can be run in the CPU.

c. File Management Unit: Responsible for coordination of the usage of a computer system's storage facilities. It maintains the records of all files stored, which users have access to them and what sections of the storage device are free so that they can be used for storing new files.

d. Network Management Unit: Responsible for coordinating sending and receiving of messages across the network.

e. I/O Device Management: This is responsible for interacting with device drivers which interpret the generic requests of the Operating System into the detailed instructions of the device. This approach allows the design of the Operating System to be independent of the devices.

f. Secondary Storage Management Unit: Responsible for working with the file manager to coordinate the utilization of all storage units within the computer system.

g. Security Management Unit: Responsible for protection of the processes and other components of the Operating system.

In higher level topics, we shall be discussing in more details the functions of each of the mentioned components of an Operating System.

Video 2: Structure and organization of Operating Systems (10 minutes)

Introduction

Welcome to the second session of this module on Introduction to Operating Systems. In this session we shall be looking at the

structure and organization of an Operating system. The structure of an Operating System refers to the various design options that are adopted when developing Operating Systems.

Structure of an Operating System

This far, we have looked at what an Operating System is and how it looks from outside. In this session, we are going to look at the different approaches that have been tried out in implementing the internals of an Operating System. This list is not exhaustive but gives some design approaches that have been implemented in the past. These design options form the basis of the structure of an Operating System and some of them include: monolithic systems, layered systems, microkernels, client-server models, virtual machines and exokernels.

1) Monolithic systems

This is one of the earliest approaches and the simplest. In this approach, the entire Operating System runs as a single program in kernel mode. Usually, an Operating System is a large piece of software comprising several procedures that have been linked together to form a single entity. In a monolithic approach, every procedure from the large pool of procedures is free to interact with each other as long as the called procedure provides some computational value required by the calling procedure.

While this may give the impression of a monolithic approach being unstructured, that is not the case. It still has some structure. In monolithic systems, the services offered by the Operating System(system calls) are requested by putting the parameters in a well-defined place such as on the stack and then executing a trap instruction. The trap instruction will make the computer switch from user mode to kernel mode, thereby transferring control to the Operating System. The Operating system will determine which service request has been made (system call that to be invoked) then fetch the provided parameters as it executes the service call.

From this organization, it can be discerned that a basic structure for an operating system comprises:

1. A main program: This is responsible for invoking the requested service procedure.
2. A set of service procedures: These are responsible for carrying out the system calls.
3. A set of utility procedures: These are helpers to the service procedures.

A graphical illustration of the monolithic architecture to be inserted here. {It should show the three components: Main procedure,

Service procedures and utility procedures. Refer to figure 1-24 in the core reference text number 1}

The greatest challenge with this approach is that it creates a scenario where we have thousands of procedures that can call each other at any time without restrictions. This results in a system that is not only difficult to understand but also easy to bring down by a bug or a crash in any of the procedures.

2) Layered Systems

This approach builds on the components identified in the monolithic approach to create an Operating System as a hierarchy of layers in which one layer is constructed upon the one below it. An example of OS built in this approach was THE system built by Technische Hogeschool Eindhoven in the Netherlands by E. W. Dijkstra (1968). It had five layers structured as follows:

Insert a graphical illustration of THE architecture based on the layers below.

Layer	Function
5	The operator
4	User programs
3	Input/output management
2	Operator-process communication
1	Memory and drum management
0	Processor allocation and multiprogramming

Microkernels

In the above approaches, all the layers were actually implemented within the kernel. However taking note that a small bug in the kernel can bring down the whole Operating System, it is therefore necessary that only as little as possible is put within the kernel. This ensures that the user processes are set to have less power so that any bug in them might not be fatal.

This resulted in a new design approach in which the Operating System is split up into small, well-defined modules, only one of which—the microkernel—runs in kernel mode. The remaining components run as relatively powerless ordinary user processes. A sample microkernel architecture is shown in the figure below:

A graphical illustration of MINIX system should be inserted here: {source: figure 1-26 of the core reference book}

Client-Server Model

	The client-server model is a variant of the microkernel concept which distinguishes two classes of processes. The server processes provide some service while the client processes utilize these services. These two classes of processes communicate with each other via message passing. The client requests for services by composing messages with the kind of service they want then send to the server process. The server process does the work then sends back the answer to the client. ***Insert a graphical illustration of the client-server model here.*** **Virtual Machines** This approach improves the previous strategies and introduces a component that runs on bare hardware and does multiprogramming, providing several virtual machines to the next layers up. This central component is what is today referred to as type1 hypervisor. ***Insert graphical illustration of virtual machine structure here.*** **Exokernels** This is a strategy in which the virtual machine is partitioned such that each user has a subset of the resources. To allow this, a special program sits at the bottom layer and its job is to allocate resources to virtual machines and check their utilization to ensure that no machine is trying to use someone else's resources. This special program is called exokernel.
ACTIVITY 2: READING READING MATERIAL 1	**Structure and organization of Operating Systems** Andrew S Tanenbaum. (2016). Modern Operating Systems Paperback. Pearson. pp 62 - 72 https://www.amazon.com/Modern-Operating-Systems-Andrew-Tanenbaum/dp/9332575770#detailBullets_feature_div

ACTIVITY 3: Comprehension questions: 	1. There are several design goals in building an operating system, for example, resource utilization, timeliness, robustness, and so on. Give an example of two design goals that may contradict one another. 2. What is the difference between kernel and user mode? Explain how having two distinct modes aids in designing an operating system. 3. What is a trap instruction? Explain its use in operating systems. 4. Explain how separation of policy and mechanism aids in building microkernel-based operating systems. 5. Virtual machines have become very popular for a variety of reasons. Nevertheless, they have some downsides. Name them. 6. What is the difference between timesharing and multiprogramming systems?
LEARNING OUTCOME 2: Conceptual knowledge ACTIVITY 4: Video to be used.	
CASE 1: 	Implementing full automation solutions in organizations usually require deployment of high end servers that run different Operating Systems depending on their purpose. These physical servers are often subject to heavy loads and run for a longer time. This at times leads to degeneration and call for an upgrade of the datacenter. Consider the case of "ABC Technologies Ltd", a company providing CCTV surveillance in a large city. They thus constantly receive lots of camera data from all over the city through their over 100 VPN tunnels into their server infrastructure which is rapidly ageing and tending toward being unable to keep with the required load. They have been advised to buy a more powerful server onto which they can implement several server environments that will be specialized to received camera data through a dedicated subset of VPN tunnels. 1) Within the context of different structure of Operating Systems, explain which structure fits the description provided in this case. 2) In order to be successful, ABC Technologies will need a hypervisor, explain what this means and give some examples that you will recommend to ABC. 3) The hypervisor mentioned above, will interact with the hardware resources through hypercalls. Compare and contrast hypercalls to System calls.

ACTIVITY 5: READING MATERIAL 	1) Virtual Machines: RedHat explanation of virtual machines https://www.redhat.com/en/topics/virtualization/what-is-a-virtual-machine 2) Explanation of virtual Machines by IBM https://www.ibm.com/topics/virtual-machines 3) Virtualization concept in details https://www.vmware.com/topics/glossary/content/virtual-machine.html 4) Explanation from Microsoft https://azure.microsoft.com/en-au/resources/cloud-computing-dictionary/what-is-a-virtual-machine/#:~:text=A%20virtual%20machine%20is%20a,on%20many%20people's%20work%20computers • Having read the above articles, you are required to write a blog in the LMS with focus on virtual Machines and Operating Systems.
ACTIVITY 6: ONLINE DISCUSSION 	Your course instructor will create a discussion forum in the LMS to facilitate online group discussions. You are required to read the discussion topic and give comments. You are also encouraged to comment on contributions from at least three members of your group. You can use the LMS platform to send questions to your instructor on the discussion topics that he/she has posted on the LMS. The group discussion will be graded based on a weight that will be indicated on the LMS.
LEARNING OUTCOME 3: PRACTICAL SKILLS VIDEO 3: 	1. Watch the 5 minute video in this link for more information about the components of an Operating System. 2. Watch the next video on the structure of operating systems. It presents information on Monolithic, layered, microkernel and other structures that you have been introduced to in the first session of this module.
ACTIVITY 7: Learner practice sessions	a) In this session, you are required to do a “lightning talk” focusing on “Evolution of Operating”. In the “lightning talk”, use your smartphone or any other video camera to record yourself in not more than “30 seconds” while explaining the “***evolution of Operating Systems***” and the pros and cons of each.

	Note: • Upload your video with the captions <*fname_lname_talk2*>. where "*fname*" is your first name and "*lname*" is your last name (or surname). The video must not be more than 30 seconds long. b) An Operating System is a very critical component of computer systems. They have evolved over a period of time with rapid advances being accelerated by changes in semiconductor electronics. • Write a brief essay to clearly outline the at least five cycles in the evolution of computer Operating Systems. a. The essay should be between one and two pages long, Times New Roman font 12. b. The essay will be submitted in the LMS under the title, "*Evolution of Operating Systems*". c. Grading will be out of 20 Marks.
ASSESSMENT OF PRACTICAL SKILL:	In the above activity, you uploaded your video, <***fname_lname_talk2***>. It will be assessed by the instructor by looking at among others: a) Accuracy of the assertions you have made (5 Marks) b) Degree of completeness of your response to the task (3 Marks) c) Adherence to the requirements with regards to topic and length of the video. (2 Marks)
LEARNING OUTCOME 4: KEY/TRANSFERABLE SKILLS	1) Operating System Structures • https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/2_Structures.html 2) Tutorials in Structure of Operating Systems • https://www.javatpoint.com/operating-system-structure#:~:text=The%20operating%20system%20is%20divided,the%20OS%20became%20progressively%20better. • https://www.geeksforgeeks.org/different-approaches-or-structures-of-operating-systems/ • https://www.tutorialspoint.com/Operating-System-Structure 3) Reinforcing the reading with a wiki • Your instructor will create for you an activity in the LMS that allows you to create a wiki as part of completion of this reading activity. • Having read about the structure of an Operating System from the above links, you are required to write a Wiki in the LMS that summarizes what you have learnt above. • Your Wiki will be graded through peer ratings on you wiki.

ACTIVITY 8	In “Activity 5”, you learnt about virtual machines in details by looking at reference materials from major providers of this technology. Use the information from those materials to create a wiki in the LMS. In your wiki, touch on the following aspects: a) Meaning of virtual Machines b) Kernel Virtual Machines (KVM) c) Importance of Virtual Machines d) Types of virtualization e) Types of Virtual Machines f) Pricing models for virtual Machines g) Compare and contrast Virtual Machines with Containers
QUIZZ: 	1. Which part of the operating system is responsible for CPU scheduling? a). Main memory manager b). I/O system manager c). System call d). None of the above 2. The two modes of operation of an operating system are called _____? a). Process and Kernel b). Kernel and user c). Interrupt and system d). Ready and running 3. An operating system module that performs context switching is called _____? a). Context switcher b). Dispatcher c). CPU scheduler d). None of these 4. Which of the following instructions should be allowed only in kernel mode? a) Disable all interrupts. b) Read the time-of-day clock. c) Set the time-of-day clock. d) Change the memory map. Answers: 1. d). None of the above 2. b). Kernel and user 3. b). Dispatcher 4. a). Disable all interrupts

	Pass Mark: 80%
TAKE HOME MESSAGE	Your course instructor will create a feedback section in the LMS to facilitate provision of your take home message. You are required to give a brief description of what you have learnt in this module in not more than half a page (typed) in the feedback section provided.
Reference list	1. Andrew S Tanenbaum. (2016). *Modern Operating Systems Paperback, 5th Edition*. Pearson. 2. Silberschatz A., Galvin P. B. and Gagne G. (2008). *Operating System Concepts*, 8th Edition. Wiley. ISBN: 9780470128725 3. Meyers, M. (2016). *CompTIA A+ Certification Guide*. McGraw-Hill Education